REMARKS

In the Office Action dated March 28, 2005, claims 1-19 were rejected under the judicially created doctrine of obviousness-type double patenting over claims 1-19 of U.S. Patent No. 6,738,896; and claims 1-19 were rejected under § 103 over U.S. Patent No. 6,182,210 (Akkary) in view of "The Metaflow Architecture," IEEE Micro (1991) (Popescu).

A terminal disclaimer is submitted herewith to obviate the double patenting rejection.

The scope of amended claim 1 has been broadened in some respects, such as substituting "numbered locations" with "locations," substituting "numbered location" with "location," and deleting the language "using the certain total number and" at line 11.

It is respectfully submitted that a *prima facie* case of obviousness has not been established with respect to claim 1 for at least the following reasons: (1) no motivation or suggestion existed to combine the teachings of Akkary and Popescu; and (2) the hypothetical combination of Akkary and Popescu does not teach or suggest all elements of the claim. *See* M.P.E.P. § 2143 (8th ed., Rev. 2), at 2100-129. As conceded by the Office Action, Akkary does not disclose calculating a value of a status bit for a selected location based on the unique number of the issued program step, and determining availability of the selected location of the execution queue based upon the determined value of the instruction valid bit and the calculated value of the status bit. 3/28/2005 Office Action at 5. The Office Action relied upon Popescu as teaching the missing elements, equating the claimed status bit with the color bit taught in Popescu. *Id.*

With respect to Akkary, the Office Action cited specifically to Figs. 2 and 24 of Akkary, and selected corresponding text passages of Akkary. Fig. 2 of Akkary is a block diagram of a processor, and Fig. 24 shows portions of a load buffer. Load instructions can be stored in the load buffer 182 of Akkary. As stated by the Office Action, the load queue of Akkary "would have to have a pointer or counter, or both to keep track of which spot is the next location in the queue that the next sequential load instruction will be stored, therefore the next sequential instruction ID, or load buffer ID, will be selected and then selecting the specific location in the queue that is the next location that will be available when a load instruction is retired in program order." 3/28/2005 Office Action at 4. It appears that the pointer or counter that the Office Action is referring to is the head and tail of the load buffer, as taught by Akkary, where the head is moved when an instruction is deallocated. Akkary, 24:25-27. This type of pointer/counter

technique of deallocating and allocating instructions to the load buffer, as taught by Akkary, refers to a queue where an instruction is deallocated from the head and an instruction is allocated to the tail of the buffer, with the pointer/counter keeping track of where the head and tail are located in the buffer. Thus, selection of an entry of the buffer to place an instruction into, as taught by Akkary, is based on the pointer/counter referred to by the Office Action (the head and tail pointer/counter of Akkary), not "based upon the unique number of the issued program step" as recited in claim 1. Note that claim 1 expressly recites selecting a specific one of a plurality of locations in an execution queue *based upon the unique number of the issued program step.* Selecting an entry in the load buffer based on head and tail pointers, as taught by Akkary, is not selecting an entry based on the unique number of the issued program step. This claim element that is missing from Akkary is not taught or suggested by Popescu. In view of the defective application of Akkary to this one element of claim 1, the *prima facie* case of obviousness is defective for at least this purpose.

Moreover, the color bit of Popescu, equated to the status bit of claim 1 by the Office Action, is used to indicate the relative age of an instruction in the DRIS. This color bit is not used to determine whether a selected location (selected based on a unique number of the issued program step) is available. Claim 1 expressly recites: based upon the determined value of the instruction valid bit *and the calculated value of the status bit*, determining *availability* of the selected location ...." The Office Action referred to a modulus operation that allows an index to wrap around in Popescu. 3/28/2005 Office Action at 6. The Office Action also referred to the queue of Popescu wrapping around to the top so that instructions can continually be sent to the load queue as the oldest instructions retire and become available. *Id.* However, the Office Action does not explain how the color bit of Popescu is used to determine whether the selected location is available, as expressly recited in claim 1. The wrap-around queue of Popescu does not use a status bit that is checked to determine availability of a selected location of the queue.

Therefore, the hypothetical combination of Akkary and Popescu does not teach or suggest all elements of the claim for this additional reason. A *prima facie* case of obviousness has thus not been established for at least this reason.

Also, there appeared to be no suggestion or motivation to combine the teachings of Akkary and Popescu. More specifically, there was no motivation or suggestion to combine the

deferred-scheduling, register-renaming instruction shelf (DRIS) of Popescu with the load buffer of Akkary. Popescu teaches the concept of instruction shelving, where out-of-order execution is used to overcome the performance loss caused by stalling. Popescu, p. 12. "The principle of out-of-order execution is that the issue logic shelves any stalled instruction and continues to issue subsequent instructions, returning to the shelved instruction only after all its operands are available." *Id.* The shelved instructions are placed in the DRIS, as depicted on page 63 of Popescu. The Office Action cited to paragraph 4 on page 64 of Popescu as purportedly teaching the elements missing from Akkary. The cited passage of Popescu refers to allocating an identifier (ID) to each shelved instruction, where the ID consists of an index into the DRIS, plus a "color" bit. The IDs of Popescu are allocated in "strict numerical order," with the color bit toggled when the index portion wraps around. The color bit of Popescu is used to determine the relative age of two shelved instructions in the DRIS.

The teaching of Popescu with respect to shelved instructions, which are stalled instructions, stored in the DRIS is completely unrelated to the load buffer, used for storing load instructions, as taught by Akkary. A person of ordinary skill in the art looking to the disparate teachings of Akkary and Popescu would not have been motivated to combine the teachings regarding the DRIS in Popescu with the teachings regarding the load buffer of Akkary.

Independent claim 10 is similarly allowable over the asserted combination of Akkary and Popescu.

Dependent claims, including newly added dependent claims 20-30, are allowable for at least the same reasons as corresponding independent claims.

Newly added independent claim 31 is also allowable over the asserted combination of Akkary and Popescu, which fail to teach or suggest the first queue and execution queue that have the recited features of claim 31. For example, the hypothetical combination of Akkary and Popescu fails to teach or suggest that locations of the execution queue store respective program steps based on queue entry numbers calculated from numeric operations on the program numbers of respective program steps. Moreover, the hypothetical combination of Akkary and Popescu also fails to disclose or suggest the first queue to determine whether a particular location of the execution queue is available by calculating a status bit based on the program number of the

respective program step, and comparing the calculated status bit with the stored status bit to determine whether the particular location is available.

In view of the foregoing, allowance of all claims is respectfully requested. The Commissioner is authorized to charge any additional fees and/or credit any overpayment to Deposit Account No. 08-2025 (200304950-2).

Respectfully submitted,

Date: 6-28-2005

Dan C. Hu
Registration No. 40,025
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024
Telephone: (713) 468-8880
Facsimile: (713) 468-8883